

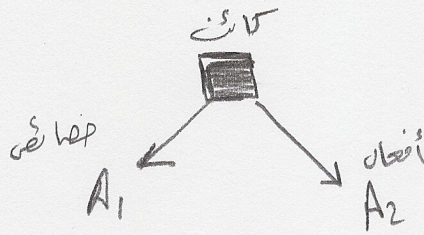
Java 101

Lesson 1

* every thing is an object كل شيء عبارة عن كائن

* every object consists of

Attributes ^{المواصفات} "property" ^{خاصة}
Actions ^{أفعال}



Example:

Car is an objects

Attributes: Model, Color ...
Actions: move-turn-fly ...

Color = red, Model = BMW

Lesson 2

Attributes: ^{المواصفات} ^{خاصة} ^{أشياء موجودة مع الكائن (أي، ينتمي إليها)}

* ^{أشياء تأتي مع كائنات} ^{قيمة خاصة} ^{اسم خاصية} Name = Value

دلالة خاصة

خاصية

age = 45

Example: ^{الصفات} ^{أشياء موجودة مع كائن}

name = Ali

age = 20

length = 180

Color = white

Lesson 2

الأفعال
Actions

يفعلها
الاشياء التي تقوم بها الكائنات

Ex

Action

object

Car

move
turn
stop

Lesson 3 :-

Car

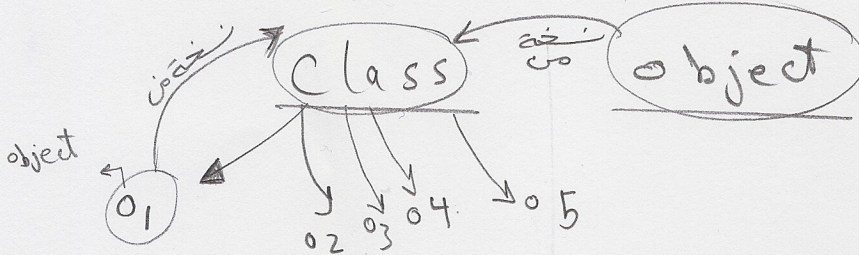
attributes	object name	Actions
A ₁	A ₂	
Color	move	
Name	turn	
Company	stop	
⋮	⋮	

« الكائنات مع الكائنات »

وهذه الكائنات من طريق جدول
أي كائنات تقدر تمثلوا بهذا الشكل
" تمثيل الكائنات "

Lesson 4 :-

مفهوم يرمز به الكائنات



* الكائنات نسخة من الكلاس

Ex: Car

وهذه السيارة

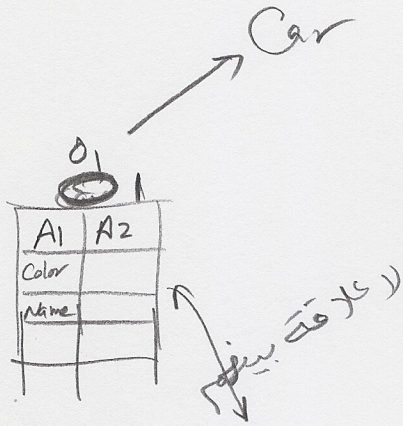
Car Class

A ₁	A ₂
Color	move
Name	turn
Model	stop

This is class
العمل فانهي بدون قيم
قيم فارغة

لا يتلقاها كائن object بل هي نسخة من جدول
ويجب بيانه

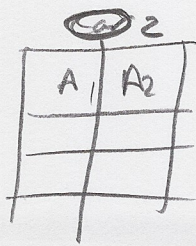
لنعريف Class اسمه Car اقله اعطيني نسخة واحدة من Car وسأسميها o1 .



وكرر هيك عمل جدول بنادرا مع ال Class الى انك عملته
نسخة منه

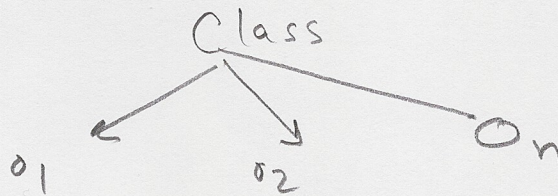
* بالخطا هونسخ جدول زي ما هوا

* وعد كر هيعي القيم



ولو قلنا اعمل نسخة اخرى اسمها o2 :
هيسروح ناسخ جدول ال Class ويحط تاي ويحي بياناته

* بدونه الكلاس لا يوجد شيء اسمه Object لان ال اجزاء هونسخة من ال Class



* الكلاس يلسأ مرة واحدة وبعد ذلك تنسخ منه ال كائنات

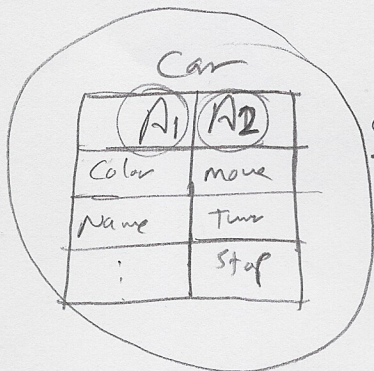
* ~~الكلاس~~ ال Object هو عبارة عن نسخة من ال Class

* Lesson 5

مفهوم ال Class

دي كلها كلامات

كلاس : ان تضع الامتياز في تصنيفات "محاد - صولة - نبات - انسان"



Class car

* لما قبي تنشأ اجزاء للكمبيوتر وهو هيك لك انالاعرف
هذا الكائن فستظهر ابي كتابت ال Class للكمبيوتر في اللفظ
عنا هعرف انواع الكائنات

عني كده لما اقله انا ده كلاس هيسروح يشوف A1 خصائصه السيارة

و A2 افعال السيارة وكره يكونه اشعرف عليها

* ال كلاس زي خريطة بالنسبة للكمبيوتر او دليل ارشادي

لوجيت على الكمبيوتر

Car O_1

عني أنشأ object جديد اسمه O_1 من الكلاس Car

عني هياخذ الجدول بتاع class وينسخو

بست كيبقي القيم بقا عني

Color = red

Model = 1991

Name = BMW

لو جيت على الكمبيوتر

Car O_2

O_2 سبب كبره

A1	A2
Black	--
...	--
2002	

O_2 & O_1 →

لا توجد علاقة بينهم

لكن

O_2 & O_2 →

لهم كلاهما مشترك

Lesson 6

البرمجة الشيئية

لغة جافا تصنف مع أنها Object oriented programming
 "oo" "oop"
 عن لغة التعامل بمفهوم الكائنات
 والكائنات

المتغير لما يكون داخل الكلاس
 يسمى خاصية

Variables

المتغيرات

attributes يتم تمثيلها في البرمجة بـ

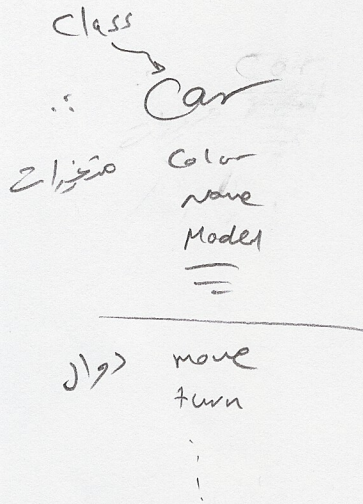
"Methods"

Functions

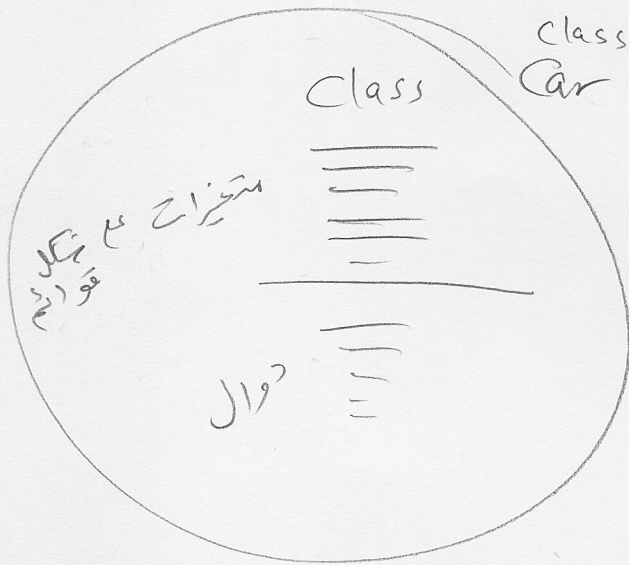
الحوال

actions

الآن لما إذا كانت
 تسمى Method
 تسمى



كل كائن سيكون بداخله متغيرات وحوال



Car x;

Car y;

← كائن Object اسمه x من الكلاس Car

كده هينسخ جميع المتغيرات والحوال وهنطوي
 فيها x

Lesson 7

حوله بلغة جافا
 Java
 الكمبيوتر
 المطلوب

لبناء class اعمل جدول
 النظري

Car	
A ₁	A ₂
Color	_____
Name	_____
////	////

Coding :-

Class car {

int model;

() {

}

Car x;

x.model = 2011

"Car"
 class
 مخصصات الclass
 مخصصات الclass

Car Z;

لو صرنا مكتبة الكمبيوتر مثلاً :-

أشياء من الكلاس car

صيني لو صارنا نورد في السيارة مثلاً "model"

Model = 2010 ← "Z" عربي "car"

Lesson 8 : netbeans ^{بيج} ^{بروژه} ^{مجاينه}

+ download program ✓

+ public class JavaCourse { ...

class _____

}

Lesson 9

Ex :-

JavaCourse.java

+ لازم كه اسم يلف بنفس اسم ار class چر وجود داره

Lesson 10

+ البته ادا جادته بداخل الكلاس ← نفس Method

+ الكلاس خريطة لكائنات

Lesson 11 : Comments

+ الكود من كائنات اليفه مطلع ~~من~~ ^{من} الكود

→ Inline Comment : ^{من كود} ^{يكونت} ^{من} ^{طروا} ^{اليفه}

Ex: // welcome to Anan

→ Multi line Comment : ^{الملاحظات} ^{تكتب} ^{في} ^{اكثر} ^{من} ^{سطر}

/* - */

note: /* *

*

@author {name}

*/

Java documentation

دليل البرنامج

Lesson 12

كيف يميز الكمبيوتر البيانات
أعزنا بالذاكرة ؟

ع/ الكمبيوتر يميز لكل خانة عنوان
عنوان يميز

و غالباً ترقيده بال hex decimal

النظام الست عشري

address
0x321
0x322
0xFc1

الرقم كبير جداً

ذاكرة الكمبيوتر

Content
10
7
35

هنا تخزن
البيانات
Size
1 byte

> 0x12FC1234DA1 - - -

In programming

Store :- "تخزين" بخر في الذاكرة

Fetch :- "جلب" بتجيب من الذاكرة

لعمري اكامل مع مكان معين في الذاكرة
لن نقرر مع العامل معه والا بالعنوان

"Store"

روح العنوان ده وعطاني القيمة 35 \Rightarrow 0xFc1 = 35

العنوان ده كبير جداً
عنه كده 4 بوا المتغيرات

المتغيرات :- اعداد مستقرة العناوين الذاكرة

Ex: X = 7 \Rightarrow

الكمبيوتر هربط X بعنوان
معين بشكل تلقائي

print x

Computer print 0x322

"Fetch"

النتج
7

Lesson 13

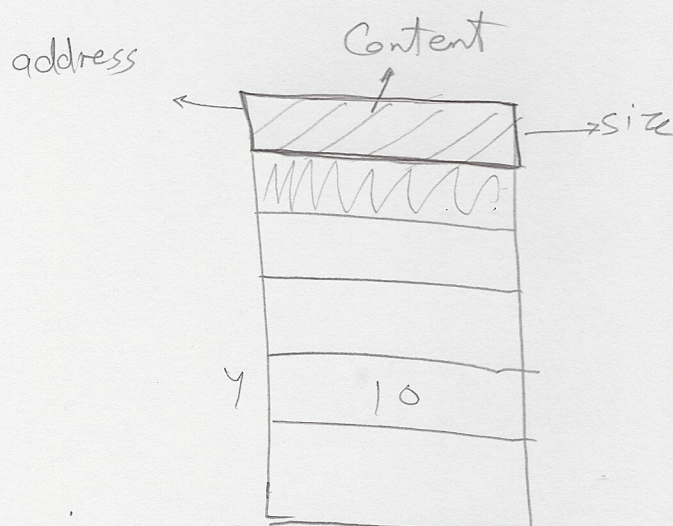
* في البرمجة : أنت تتعامل مع الذاكرة عن طريق المتغيرات

= لا تقدر الوصول للذاكرة بطريقة مباشرة ، لأنك لست تعرف المتغيرات .

* المتغيرات : وسيلة يستخدم المبرمج للتخزين في الذاكرة ، والاستفادة منها .

Constraints
بالإمكان وضع قيود على الـ cell تخزن فيها
نص - رقم - عدد بـ عشرة عشرية - ...

Constraints: حدد نوع القيمة ليتمكن البرنامج
ادخل الـ cell



* Data types

Char : حرف character	نوع البيانات String	نص
	int	رقم صحيح
	float	
	double	

Ex : int x : الخلية x لا تقبل إلا عدد صحيح

Char z : كده معرفت متغير اسمه z
ولازم يكون محتواه حرف "char"

int y = 10 : حجز مكان في الذاكرة اسمه y
ولازم هذه الخلية تكون عدد صحيح

data type	name	=	value
النوع	الاسم		القيمة المتغيرة

قيمة المتغير في الغالب يجب أن تكون موافقة لنوع البيانات

Lesson 14

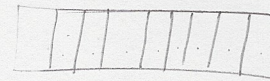
type name = value

Size يختلف من نوع Type الى نوع

Ex: `int x = 10`

كل نوع له حجمه في الذاكرة

1 cell = 1 byte = 8 bit



* `int` → 4 cells → 32 bit
 ↓
 الخلية الواحدة

* `float` : عدد عشري "عدد بفاصلة عشرية" 4 byte

* `double` : عدد بفاصلة عشرية أطول 8 byte

Lesson 15

Numerical Data type
 أنواع البيانات العددية

0 → 255
 byte = 256

In java :-

byte

-128

127



2 byte

short

عدد صحيح

int

عدد صحيح أدق

long

عدد صحيح أدق

float

عدد بفاصلة عشرية

double

عدد بفاصلة عشرية أدق

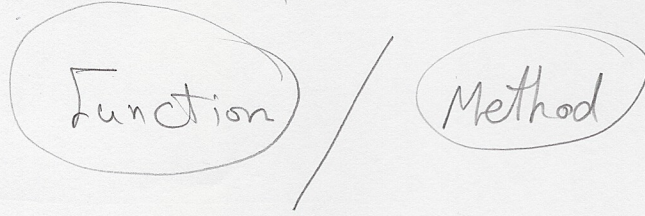
Lesson 16 :-

مفاهيم من المتغيرات

* Declare variable " public int model ;
 اسم المتغير ← نوع البيانات ← يكتبها كده دلوقت وبعدين هتفهمها

Lesson 17 :-

مفهوم الدوال



والدوال هي Function
 بيت الاسماء فقط بيستخدموها كده

Class {

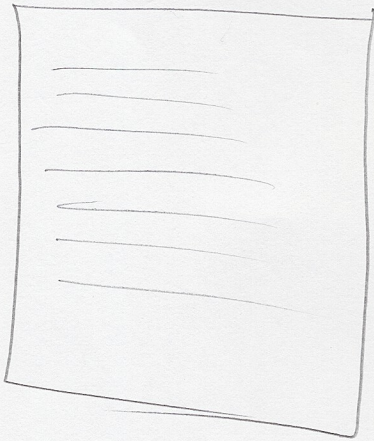
Function

البراق اذا جاردت بداخل الكلاس
 تسمى Method

}

ان كانت له خاصية فاعال

ملف برجي



Actions
 تنفيذ
 Functions

له اسم وبيادله كود برجي



Function
 البراق عبارة عن Block
 ↓
 Function definition
 تعريف الدالة

* البلاك الذي يحتوي على السطر لاسم تنفيذه، ولا اذا تم استدعائه

لو عاوز استدعي الدالة get نكتب

get

get()

استدعاء الدالة Function Call

روح للبلوك رالي اسم get ونفذ الشفرة التي بداخله

Lesson 18

تعريف الدوال

public void Move()

{
 //
}

استدعاء الدوال

Lesson 19

Function definition

التعريف يكتب مرة واحدة

Function call

والا استدعاء يكتب مراراً وتكراراً

استدعاء الدالة

stop()

Lesson 20

معلومات إضافية عن Class

03

02

object 01

Class X

} attributes

لا يوجد كود يكتب
خارج الدوال "البلوكات"

ملاحظة: لا نضع استدعاء الدالة خارج الكلاس

Lesson 21

مثال بسيط

`System.out.print ("The value of c is" + c + " Kilogram");`

Capital

استخدم علامة زائد لوترتيب المتغيرات
للربط

Lesson 22

إشارة دالة مبسطة للتوليف

`public void speed ()`

`{ int a, b, c ;`

`a = 10 ;`
`b = 40 ;`
`c = a + b ;`

`System.out.print (c) ;`

`}`

`int a, b, c ; = int a ;`
`int b ;`
`int c ;`

لأن سيع تنفذ هذا الجزء
رأينا استدعاء
`speed()`

Lesson 23

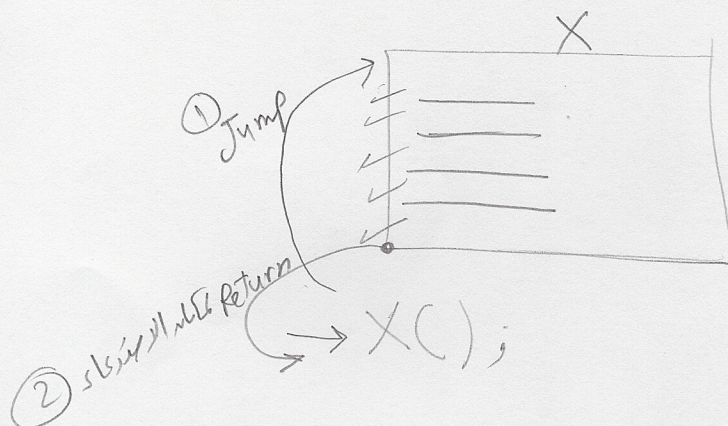
نوع الدالة

`type name ()`

`{`

`}`

لا تكتب الرأية تكتب هذا الشكل




```

type name ( )
{
}

```

ex: int nam ()

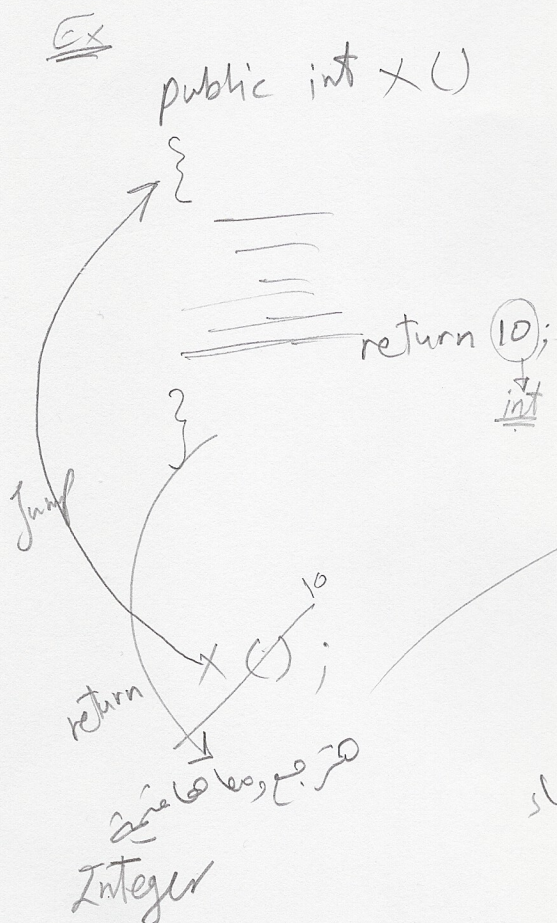
```

{
}

```

معنى النوع عندما يأتي قبل اسم الدالة

ج / الدالة لا ترجع وقت الـ return هتارجع ومساها عدد integer



الدالة x لما نتبع استدعائها
هكذا الشكل:

x () ;

معنى

return

روح ارجع مكانه الاستدعاء وامسح الاستدعاء

وعوضها بالشيء الذي

Lesson 25

الصفة العامة

```
public type name()
{
}
```

تجيب على return

```
public int speed()
{
    int a = 10;
    int b = 20;
    int c = a + b;
    return c;
}
```

في return معناها - ارجع القيمة الاسمي وعوض مكانه بالقيمة

Lesson 26

```
public void welcom()
{
}
```

نوع الدالة void

* لا يتم وضع دوال النوع void في عملية الحساب

void =

الدالة لن ترجع قيمته وقت الرجوع
حينها لا تستخدم الدالة مثل هل ترجع قيمة
عنه كذا لا يستخدمونها return

Lesson 27

مفهوم ال parameters

argument

parameters: عبارة عن متغيرات يتم وضعها بين أقواس الدالة ويتم التحويل عن قيم هذه المتغيرات أثناء استدعاء الدالة

Sum(10,12) حينها - متغيرات تظهر للدالة كما هيل يتم التحويل عن قيمها أثناء الاستدعاء

ملحوظة: - أي مجهول يحتاجه ال action يمثل ال parameter

Lesson 28

```
public class Car {
    public int model;
    public void speed(int s) {
        System.out.print("speed is " + s);
    }
}
```

مثال مع Method

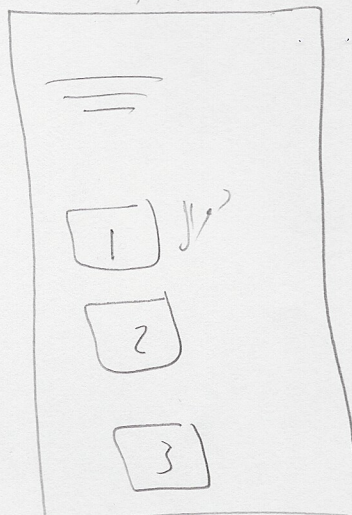
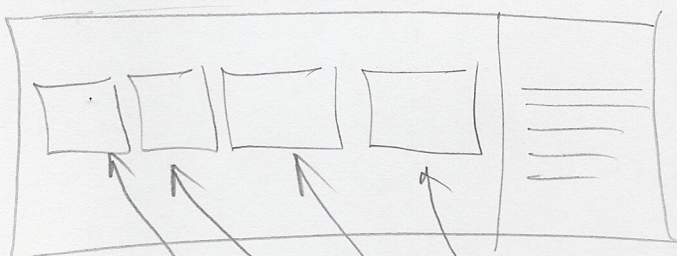
Lesson 29

معلومات أكثر عن ال Class

مم

O₁ O₁

Class X



X O₁; \Rightarrow المستطيلات

هذه نسخ المتغيرات

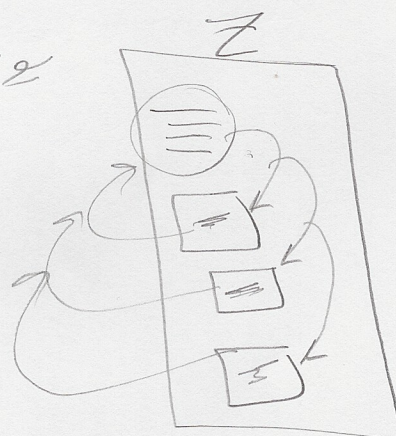
والسؤال هيتم استدعائها من ال class الاساسي
بدل ما ننسخها.

الاختيار مشترك وظل مكانها والغير مشترك يتم نسخها.

Lesson 30

في حالاته ترتبط بين المتغيرات والسؤال

action: \rightarrow تغيير قيم المتغيرات



بعض الروال تقدم بتغيير قيم المتغيرات
او الرجاء

Lesson 31

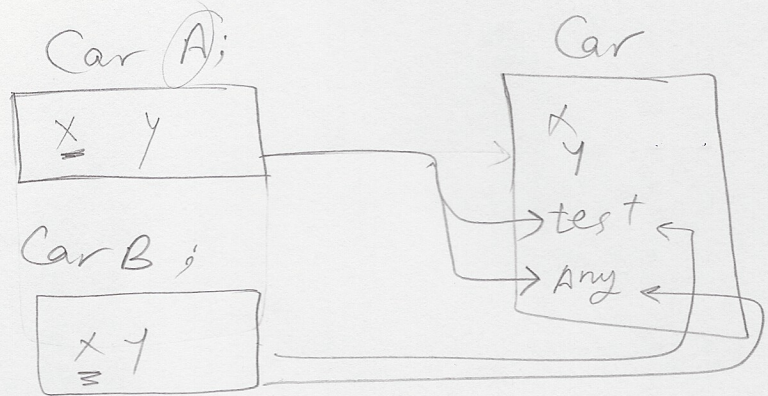
this متغير استغیر

this: متغير للوصول لخصائص الـ Class داخلياً
this: لا تستخدم خارج الكلاس

Lesson 32

this

بر/نوع
عنه نسخ: instance



⊗ هنا الفرق بين x الموجودة في A و x الموجودة في B

▲ this:

← متغير يتم استدعاه وقت انشاء الكائن
← باسم الكائن

+ this: متغير مجهول سيحل الكائن الذي يتم انشاءه

Ex:

```
public class Car {  
    public int x;  
    public int y;  
    public void test()  
    {
```

`this.x = 10;`

`this.y = 500;`

`}`

```
    public void any()  
    {
```

`this.test();`

`}`

`}`

In case of Car A:

`A.x = 10;`

`A.y = 500;`

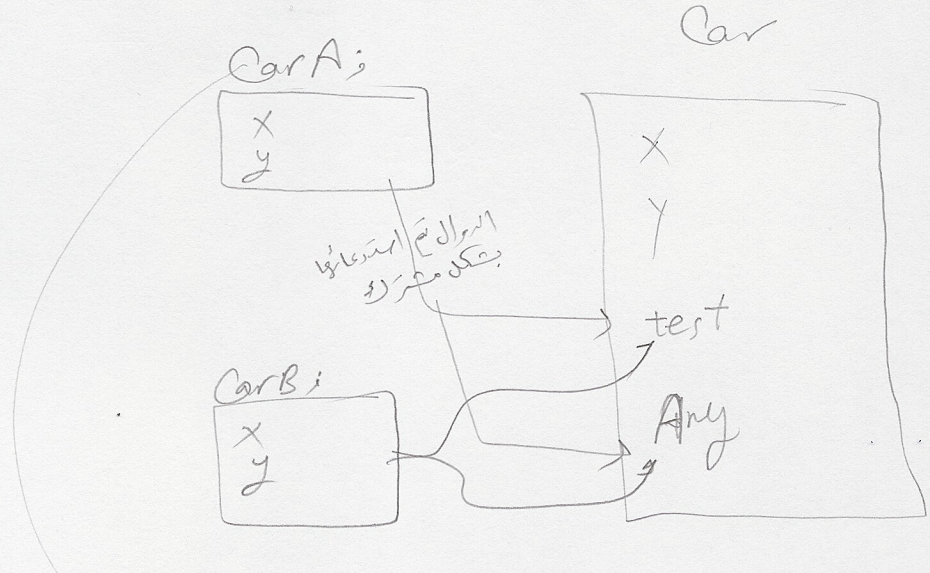
In case of Car B:

`B.x = 10;`

`B.y = 500;`

Lesson 33

المراد مع this



س/ كيف نعرف ان test نستخدم مع الكائن A أو الكائن B ؟
ج/ باستخدام متغير this

this.test() ; تكون A.test
معناها: طبق الدالة test مع متغير الكائن A

Lesson 34

مثال

Lesson 35

1 إنشاء كائن : new

Car Camry = new Car() ;
أكتب اسم الكلاس وكائن الدالة
⇒ object Camry from the class "car"

new Car :- إنشاء نسخة جديدة من الكلاس car وضربها في الكائن Camry

~~طريقة أخرى لكائن كائن~~
~~Car Camry~~
~~طريقة~~

* دالة ال Constructor *

```
public Car ()
{
}
}
```

Constructor *

* اسم دالة ال Constructor عادة تعرف بـ "تميزها" ← هيكون اسمها بنفس اسم الكلاس

* لا يوجد نوع إرجاع ~~void~~ لأنها تلقائياً ترجع object .

* وظيفتها :-

* تسترعى وقت إنشاء الكائن .

⚡ ملحوظة :- إذا لم نضع كود دالة ال Constructor في الكلاس ← سيتم استيعاب Constructor افتراضياً بواسطة Java

* يستخدم لتعويض القيم : حين أول ما يتم إنشاء الكائن ← أثناء لحظة الإنشاء

ماذا تريد أن تفعل ؟!

مثلاً : $this.speed = 0$; حينها $speed$ = 0
استدعاء ال object .

Lesson 37 :-

Constructor & parameters

Car.java :-

```
public Car (int s)
{
    this.speed = s ;
}
}
```

Java Course . Java

Car Camry = new Car (60) ;

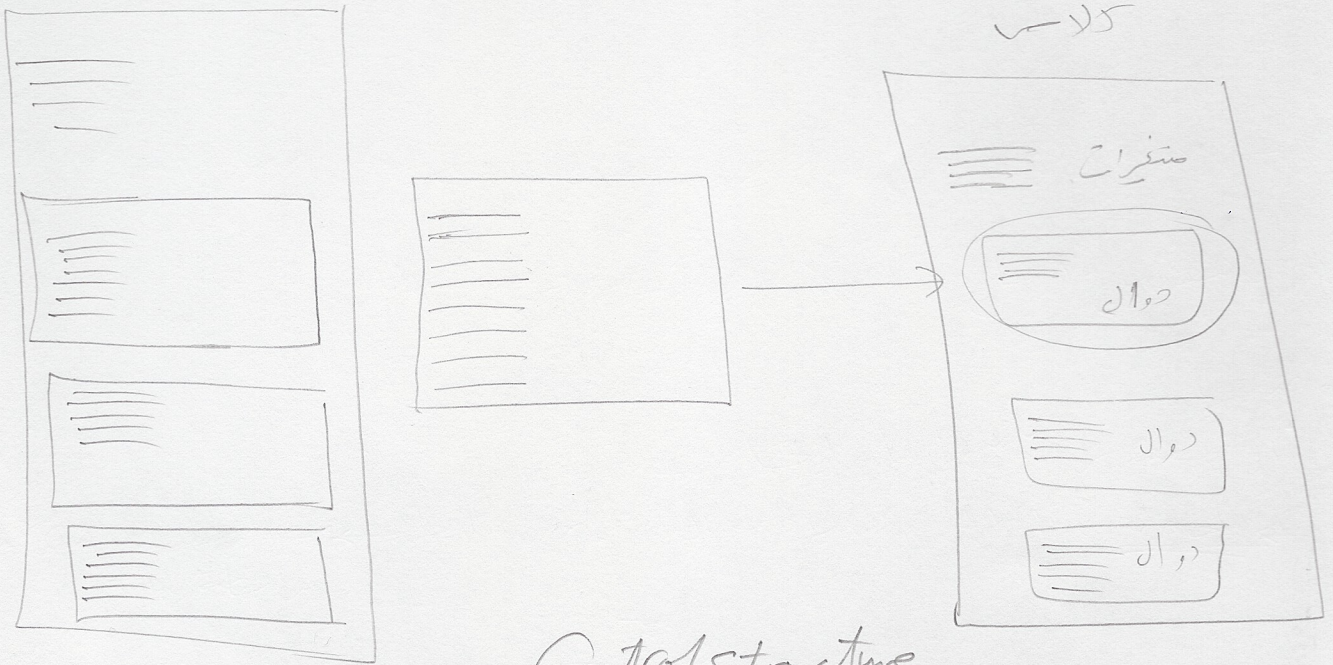
* دالة ال Constructor بها انما دالة
يمكن ان تأخذ parameters

Lesson 38 :-

مکان بیضا مع کلاسیک
والطباعة print

Lesson 39 :-

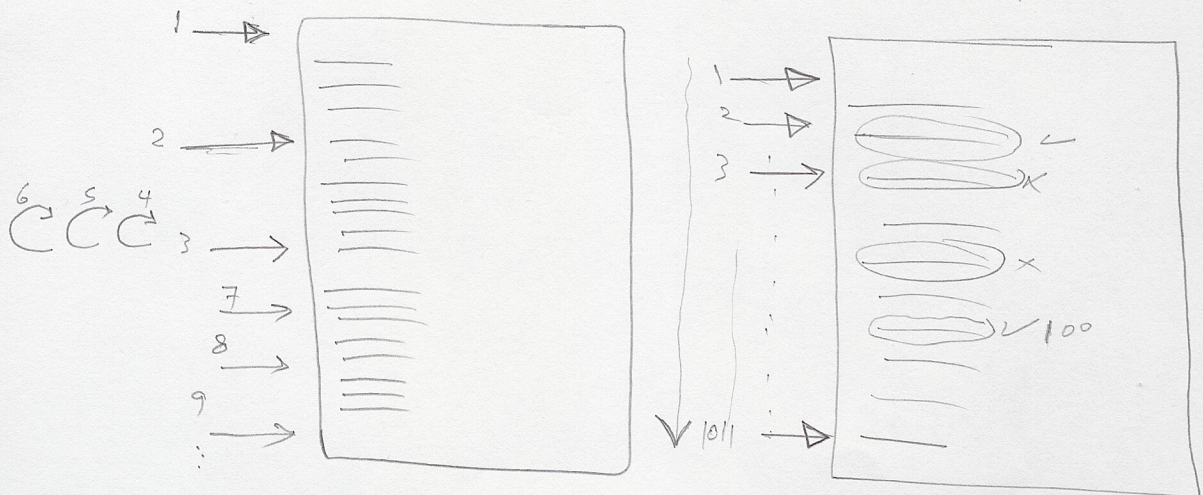
ملاحظ حول الكائنات والكلاسات



Control structure

Lesson 40

مفهوم التحكم
Control Structure



التحكم في سير التنفيذ "أوامر التنفيذ" : Control Structure

أوامر التنفيذ وأوامر التنفيذ
أوامر التكرار وأوامر التكرار

If (شرط)

{

=====

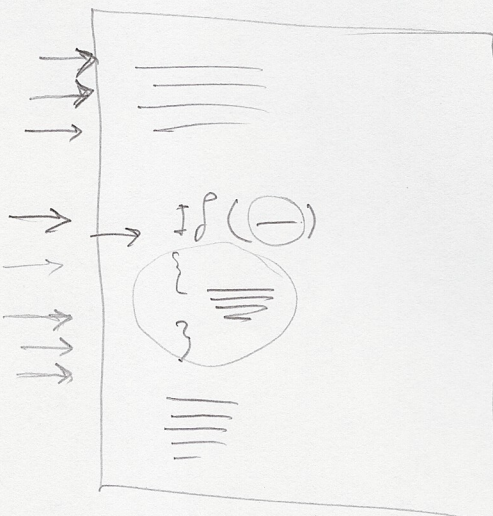
}

true

لو الشرط صحيح فلينفذ ما بين الأقواس

و لو لم يتحقق الشرط فليجاء إلى

مراحل قفز لبعد If كلها.



Ex: int a = 5;

If (a > 10)

{

=====

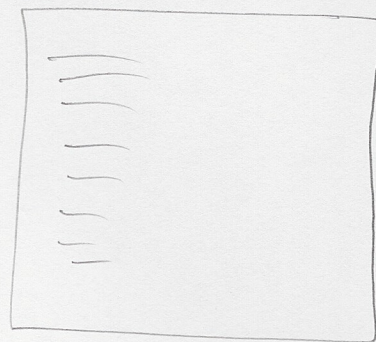
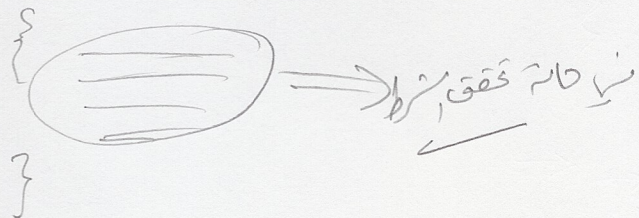
}

كده كبيره كده يكون ده
بنا لم يتحقق يكون

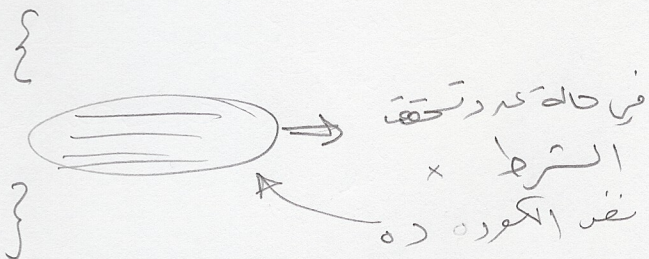
Lesson 43

استخدام الالة else

الشروط if



else



Lesson 44

مقالع else

Lesson 45

استخدام والمقارنات في جافا

أكثر 100 >

أقل <

يساوي ==

لا يساوي !=

أكبر من أو يساوي >=

أصغر من أو يساوي <=

دي المساواة الرتيبة

200 > 100

200 > 100

5 = 5

b2.equals(b3)

!b2.equals(b3)

يعني أو

يعني و

يعني لا يساوي نفسها

المساواة حرفية

لها درجتها بعدد

Lesson 46

else if أو إذا كانت

if

{

else if

{

else if

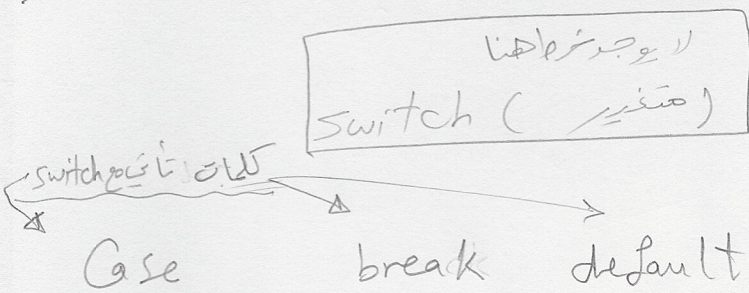
else

{

لو لم ينفذ ده
علا إذا شئت على ده

و يمكن بدون else

switch



ملاحظة: لنرم نضع Break; للخروج والا سيستم
بمتنفيذ الكود الآخر

قيمة
اختبر المتغير الموجود بين القوسين
هل هي قيمة X

switch (x)

{

Case 1:

==

break;

لو نفذ الكود هذا
ثم اخرج للنجاح

Case 70:

==

break;

لم يبقها
هل هي 70؟
اذا 70 نفذ الكود

}

استخدام switch مع أكثر من Case

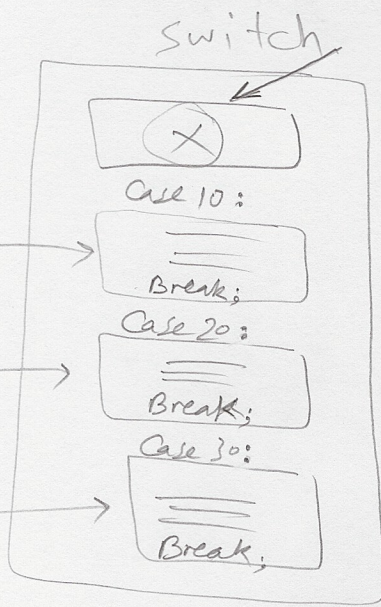
شكلها قيمته X

لو كانت 10 نفذ الكود ده

لو كانت 20 نفذ الكود ده

لو كانت 30 نفذ الكود ده

عني سيتم تنفيذ بلوك ما هو فقط
وليس كلها



دي الحالة المتسلسلة

switch(x)

```
{  
  Case 10 :  
  Case 20 : } →
```

إذا كان في أي
نقطة البلوك هذا

```
break ;
```

Case 30 :

```
break ;
```

```
}
```

Case : تبدأ البلوك

break : تنهي البلوك

Lesson 49

مثال

Lesson 50

استخدام Default

مثال

عندما لا يوجد فرع يتوافق مع المطالب يتنفذ الـ default

Lesson 51

استخدام أكثر من Case

Lesson 52

استخدام Inline-If

عبارة عن If ثنائية ولكن مختصرة.

```
if (s=10)
    system.out.println("up");
else
    system.out.println("Down");
```

int c=10; else

int y = (c==10) ? 50 : 70

الشرط لو تحقق الشرط لو لم يتحقق الشرط

لو c=10 يبقى الشرط تحقق إذا y=50
لو c≠10 " لم يتحقق " y=70

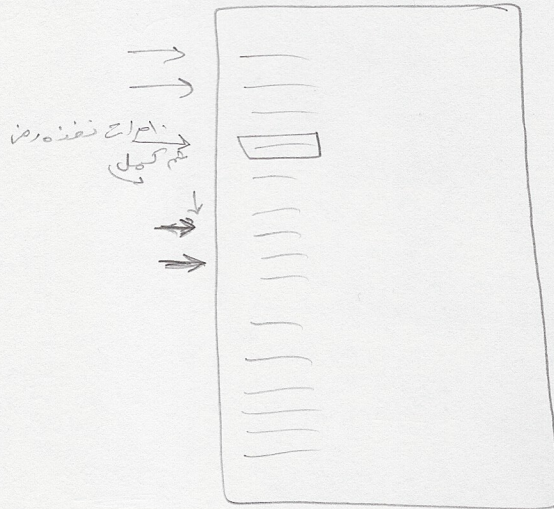
ويمكن شملها بأخذ شرط

Lesson 53

مفهوم loop / الحلقة / التكرار

وظيفة: تكرار أمر معين عدد محدد من المرات.

أو تكرار مجموعة من الأوامر عدد معين من المرات.



Lesson 54 :- & Lesson 55

عداد
زيادة, النقصان
Initialization
Increment, decrement part
الشروط
for (int i = 1 ; i < 10 ; i ++)
 {
 استمر في اللولب طالما أن $i < 10$
 زيادة بواحد
 or $i = i + 1$
 }

Lesson 56

مفهوم الزيادة في For
for (int i = 10 ; i >= 1 ; i --)
 {
 System
 }

Lesson 57

معلوم في الـ for
 For آلي عمل
 (

Lesson 58

infinite loop لولب غير منتهية
 $int\ i = 0;$
for (; ;)
 {
 لازم تحط شرط
 معين للخروج $\rightarrow i \leq 10$
 else
 لازم تضغط CP $break;$ لكسر اللولب
 لا يتم عمل Infinite $i++;$

Lesson 59 :-

معلومات عامة عن For

Lesson 60 :-

ما هي مهام عن For
أستخدم loop داخل loop

Lesson 61 :-

طباع loop بشكل متتالي

Lesson 62 :-

طالما الشرط متحقق
يتم تنفيذ الكود

while

loop

int x = 0
while ()

{

++i
}

عشاهة تخرج
ولا تكون Infinit

* يعني loop انها شرط فقط

والملومات الأخرى مسؤلية ليدرك أنت

و يتم تنفيذ الكود طالما أنت الشرط متحقق

يعني لازم تكمل ++i عشاهة تخرج

Lesson 63 :-

ما هي معلومات عامة عن

do while

ال while زي ال if بالضبط

ولكن ال if لن يرجع مرة أخرى لفوق

بينما ال while يرجع لفوق عندما الشرط لا يتحقق

Lesson 64

do while

* أكتي while معلوم

* while يتكلم بعد الأوامر في فرايز ولا في معلوم
معلومة while ;

int i = 1 ;

do

{ System.out.println(i)

} ++i

while (i < 4);

الفرق بينها وبين () while

أن ال Do while ستنفذ الكود مرة واحدة مع الأقل
عشاهة الشرط يكون بعد الكود

Lesson 65

Java loop

خلاصة

for (; ;) { }

إذا كانت النهاية معلومة

while () { }

غير معلومة

do { } while ();

نزي ال while وإذا استخدم
الكود مرة واحدة مع الأقل

Lesson 66

أنواع البيانات

Char c = 'z';

لازم يتم وضعه بين تنقيط مفرد

حرف واحد فقط
لا حرف واحد فقط

Lesson 67

String

String name = "Hi!";
System.out.println(name);

جافا عند class خاص بال String مسبق مع الفيز
وبالتالي له دوال ومميزات يمكن استخدامها

Lesson 68

تابع دالة CharAt ()

Cx String name = "Anan";

Char c = name.charAt(1);

System.out.println(c);

كله هيطبع حرف n

ملاحظة هامة :-

في حالة ال String

يبدأ العد من الصفر

وظيفة الدالة هي طباعة حرف معين في مصفوفة

Lesson 69

String تابع

length

طول السلسلة

name.length();

النتيجة int

Lesson 70

name.length() مع CharAt()

Lesson 71

الشيء الجديد بالإنجليزية

String value = new String("anan");

Lesson 72

الشيء الجديد مع String uppercase

على كائن السلسلة حرفه كابتال

name.toUpperCase();

Lesson 73

toLowerCase();

Lesson 74

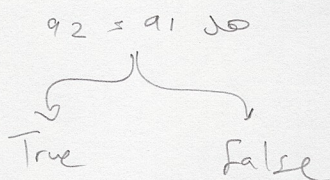
دالة مقارنة السلسلة

String a1 = "aaa";

String a2 = "aaa";

a1.equals(a2);

مطابق، true مع string



Lesson 75

equal & Ignore case

عین کر رہا تھا وہم الحروف کلمہ یا small

عین "AAa" = "aaA" کا دی

Lesson 76

Array / مصفوفات

نوع
int x = 10; → (x) → []

مصفوفة
نوع
int Z[]; → (Z) → []
0
1
2

المصفوفة :- عبارة عن متغير اكثر من قيم "يتوي مع اكثر من قيمة"
فيكون هو Index عین کہ میہ تا اہر قسم بدایہ من ہر
* الاول للقيمة في مصفوفة :-

دہ کرہ متغیر
Z[0] → مصفوفة Z
وہات القيمة اليها في المصفوفة

لا تكتب مربعات ما هي عین کرہ انك بتعرف مصفوفة

Z[]

Lesson 77

طريقة كتابة المصفوفات

$\text{int } z[] = \text{new int}[10];$

منشئ مصفوفة
مستوى 10 قيم

1- جزئي 10 خانة
للمصفوفة

Lesson 78

ادخال القيم للمصفوفة

$\text{int } z[] = \text{new int}[4];$

$z[0] = 1;$

$z[1] = 50;$

$z[2] = 300;$

$z[3] = 1000;$

index
فهرس

تم العنقود عند 3 لان الكد يبدأ من 0
يجب يوجد 4 أماكن من أول فهرس

Lesson 79

«تعبير لمصفوفة»

طريقة منحصر كتابة المصفوفات وادخال القيم

$\text{int } z[] = \{1, 50, 300, 1000\};$

ولكن يجب ان تكون القيم كـ Integer او double مع حسب اي اناست معرفه

دمك string ايضا ← عادي

د ټولنو د ژوند

int C] r; ≡ int C] r, Z;

```

int r, zc]
int asj iro ←
int ysj iro ←

```

Lesson 81

$\frac{81}{\text{int}}$ $\xrightarrow{\text{int}}$ گروه نام ۱، ۲، ۳
 لازم است ۱، ۲، ۳ گروه ۱، ۲، ۳ نام ۱، ۲، ۳
 $\text{int values}[] = \{1, 2, 3\}$

Lesson 82

values: length \rightarrow size

مع معامله المفقوده كذا خا كانت باطله دوان

Hint For C int i = 0 ; — ; —)
Index { 0 1 2 } ← (المناظر)
{ 0 1 2 } ← (مواقع)

Method
ZuB
ZuL

Lesson 83

der, der zugeht, der zu, von

```
String name = "anan";
```

$$\text{int } Z[] = \{1, 2, 3, 4\}$$

```
System.out.println(name.length());
```

دالة "يستخرج منها الأقوال"

```
System.out.println (arr . length);
```

→ خاص ہے
فیضانِ وجودِ اقدس (۱)

صفوف لفظی

```
String values[] = { "anan", "Sara",  
                    "Mido",  
                    "anan"  
                    };
```

الترتيب ده حتميا فقط
شكل عن المبرمجين

```
System.out.println(values[0]);
```

← کرده طبع anan

* كل قيمة في صفوف حاداً كانت نصية أو عددية تعتبر كأنها متغير واحد "متغير بقيمة"
عني values[0] يعتبر متغير يشير إلى القيمة "anan"
عني اسم الصفوف مع رقم الاندكس يمثل اسم متغير طالع مثل
String name = "anan";

main

جميع صفوف ثابت أو متغير

٢٤

```
int values[] = new int[10]; // fixed size array
```

2:40
main

```
public int[] createArray(int size) {  
    int c[] = new int[size]; // variable length array  
    return c;  
}
```

نوع الإرجاع: صفوف نوعها int

لو لم يتم وضع [] بجوار int سيظهر خطأ لأن c صفوف نوعها int و لو لم
تكون المتغير واحد بين [] int لعدة عناصر

Lesson 86

استخدام for each للمصفوفات

* تعتبر for each لوحة loop خاصة للتعامل مع المصفوفات

int values[] = {1, 4, 9};

* نفس for العادية لكنه يعد لوحة
غير مع الجمع عناصر المصفوفة.
كده هيسر لوحة

for (int val : value) // for each

متغير

الجمع مصفوفة
بمكون أقواس

```
{
    System.out.println(x);
}
```

* يعني : كل مرة تأخذ لفظة واحدة من المصفوفة values يعني هذا العنصر
1 وخرجه في val

ثم في اللفظة الثانية هذا العنصر 4 وخرجه في val

+ يعني يا فتها for each فكرتها انك بتعرف متغير
وهي تعرف بانك الـ values مصفوفة فبالتالي تعرف طولها
فأنت لا تحتاج تعرف طولها كرام

(اللفظة بتخزين الـ values القيمة في val)

Lesson 87

مصفوفة اسمي aa

aa.printArray()

وليست
aa[99]

تصدير مصفوفات الدوال

كباراميتر

اسم المصفوفة يُمرر للدالة بعبارة أقواس []
انظر الفيديو للفهم

Lesson 88

هل for each تُغير القيم؟

هامة

* for each لا تغير من قيم المصفوفة ولكنها تأخذ نسخة وقد عملت عليك براهين
ولكن القيم الأصلية لا يتغير عند تعديل عليها

Lesson 89

مصفوفة الكائنات

+ عمل مصفوفة من الكائنات طبيعي مثل عمل أي نوع آخر

~~Car~~ $y[] = \text{new Car}[10];$

+ أنتريد مصفوفة

اسم $y[]$ من الكلاس Car
وهو مكون من 10 كائنات من Car

خاتمة الامتحان

$y[0].\text{setSpeed}(100);$

$y[1].\text{setSpeed}(200);$

⋮

لكن كده غير صحيح
فيجب تعريف عناصر المصفوفة، إنشاء

$y[0] = \text{new Car}();$

$y[1] = \text{new Car}();$

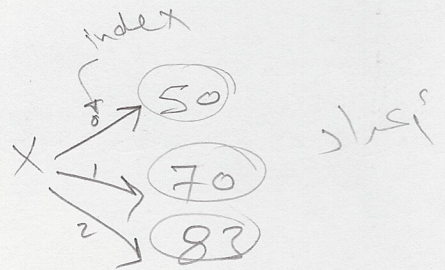
Lesson 90

المصفوفات متعددة الأبعاد

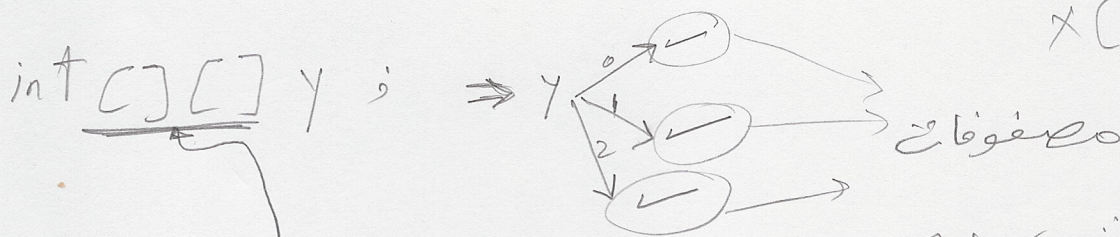
Multi-dimensional array "2D"

Ex:

$\text{int}[] x = \text{new int}[3];$

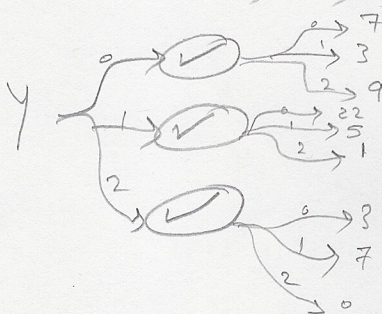


$x[0] \Rightarrow 50$



+ يعني الفهرست نفسه عبارة عن مصفوفة وليس عدد لأن 2D

يعني يعني يعني



أنشأ مصفوفة نوع int

وكان عنصر فيها عبارة عن مصفوفة
بنوع int

Ex: $\text{int} \llbracket \rrbracket \llbracket \rrbracket \gamma = \{ \{1,2\}, \{3,4\} \}$

قوس المصفوفة الأولى

التعريف 2D :-

* المصنوفة الكبرى عبارة عن عنصريين وكل عنصر عبارة عن مصنوفة أخرى
* هي عبارة عن مصنوفة كل عنصر فيها عبارة عن مصنوفة أخرى.


توضیح علی الصفحات 20

Lesson 91 :-

int m [] [] = { { 1, 5, 6 }, { 5, 7, 9 }, { 3, 1, 5 } };

\swarrow \searrow
 rows cols

rows col.



$\text{int } m[][] = \begin{cases} \text{الصف 0} \rightarrow \{1, 5, 6\}, // \text{الصف 0} \\ 1 \rightarrow \{5, 7, 9\}, // \text{الصف 1} \\ 2 \rightarrow \{3, 1, 5\} // \text{الصف 2} \end{cases}$

اشادہ مصروفہ منقذہ باستخدام new

Lesson 92
EX

Ex
int y [] [] = new int [2][4];

$$\begin{aligned} y[0][0] &= 3; \\ y[0][1] &= 5; \\ y[0][2] &= 13; \\ y[0][3] &= 15; \end{aligned}$$
$$\begin{aligned} Y[1][0] &= 9; \\ Y[1][1] &= 8; \\ Y[1][2] &= 7; \\ Y[1][3] &= 6; \end{aligned}$$

Lesson 93

اللووب الصفوفية بقدره

* المبرمجين يستخدموا اثنين لوب للعامل مع الصفوفية السابقة

$\text{int } x[][] = \left\{ \begin{array}{l} \rightarrow \{11, 53, 66\}, \\ \rightarrow \{9, 71, 63\}, \\ \rightarrow \{7, 68, 69\} \end{array} \right\};$
تتضمن الصفوفية

for (int i = 0 ; i < x.length ; i++)

{ int z[] = x[i];

for (int y = 0 ; y < z.length ; y++)

{ system.out.println(z[y]);

}

}

كده يطبع كل عناصر الصفوفية السابقة

Lesson 94

مثال للووب الصفوفية بقدره
مثال مبسط عن السابقة

كلام جمل للفرم

Lesson 95

استخدام Float و Double

float

هذه من تحت أنواع البيانات Datatypes

float x = 1.2f

له عدد بفاصل عشري
"4 byte"

Double y = 1.2543211112

له عدد بفاصل عشري أدق
و لكن حجمه أكبر "8 byte"
في الذاكرة مثال float

يتم كتابة f في حالة float لأن الاقتران في الأرقام العشرية
هو ال double لذلك يتم إخبار الكمبيوتر بأن عدد float يرفع
ألف f بعد الرقم.

Lesson 96

استخدام short و long

byte		في الذاكرة
short	→ 16 bit	= 2 byte
int	→ 32 bit	= 4 byte
long	→ 64 bit	= 8 byte

كل
أعداد
نوعية

القيم ما بين السالب والموجب

Lesson 97

استخدام النوع Boolean

Boolean x ;

عبارة عن متغير يأخذ قيمتين:

true	or	false
صح	or	خطأ
1	or	0

غير ينفع في الشيء الذي به حاليته

"أما أو"

+ ينفع في الشروط مثلا / ذكر - أنثى / حي - ميت .

Lesson 98

استخدام النوع Byte

byte → 127 إلى -128 من عدد صحيح يأخذ

ومعلوم بأن البايته في الكمبيوتر 256 خانة تبدأ من صفره وتنتهي عند 255

+ دائما يستخدم في البرمجيات التي تتعامل مع الحاردوير

+ يستخدم في التشفير و صناعة الفيديوهات

+ وفي البرمجيات التي تتعامل مع الذاكرة

Lesson 99:-

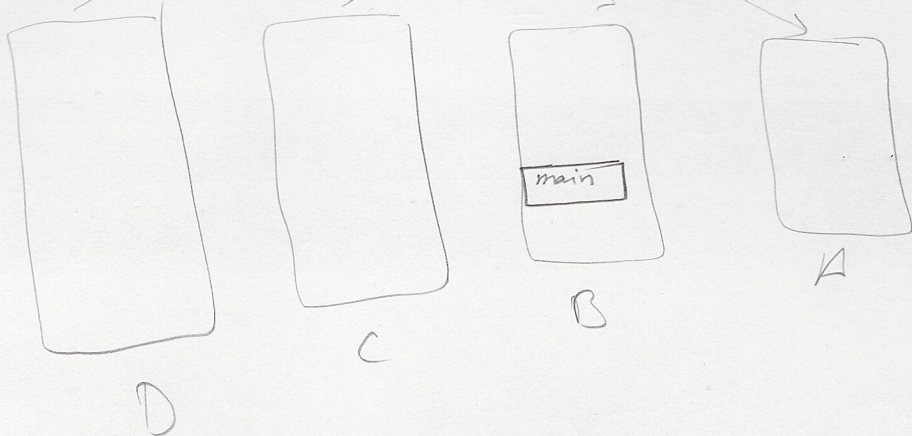
توابع مهم



Lesson 100:-



معلومات عن دالة main



الدالة لا يمكن
أن يكون تعريف
في استدعاء

* عند الضغط Run يبدأ تشغيل البرنامج
تقوم جافا بالبحث عن دالة `main()`
وأول ما جافا تلمس `main` سيبدأ التنفيذ فوراً من عندها

* تسمى دالة ال `main` ← `entry point` نقطة البدء/ نقطة الدخول
* جافا أول ما تلمسها تدخل تنفيذها هي ال `main` وآخر شيء هو جافا تخرجها هي ال `main`

Lesson 101:-

مثال على الجمع

* شاهد الفيديو

* بإمكاننا إنشاء كائن من كلاس بنفس الكلاس "يعني داخله".

مثال على إنشاء كلاس Calc

Lesson 102

* عند إنشاء مشروع جافا أنت بتعمل مجموعة ملفات جافا وتستعملها من طريق
تعريفها بداخل بعضها.

* طريقة التوابع بين ملفات المشروع ← عن طريق تعريف كائنات بداخل الكلاسات بعضها لبعض

Lesson 103 :-

مثال على خاصية دالة

Lesson 104 :-

مثال على التعامل مع التكرار

site.length() - 1
→ علامة العدد يبدأ من صف

Lesson 105 :-

مثال تحويل بعض اى معلومات

Lesson 106 :-

خرب عنام معلومات

* لانهم تكون معلومات في النوع حتى ٢x٢ أو ٣x٣ " تحت الطول "

Lesson 107 :-

عرف مشكلة خرب المعلومات

Lesson 108 :-

مثال لكلاس اكتاب

Lesson 109 :-

* برمجة الكائنات تعتمد على الوهم الدقيق .

* بالإمكان تعريف أي كائن داخلي أي كلاس
وكذلك استخدامه داخله وصغيره .

Lesson 110 :-

* هذا الجزء من الدورة كانه مدرج كيفية بناء

Lesson 111 :-

المجلد وقراءتها Syntax

Lesson 112 :-

Lesson 113 :-

Lesson 114 :-

Lesson 115

Lesson 116

Lesson 117

Done